



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/709,433	11/13/2000	Jeff Stewart	424992000200	4323

23550 7590 11/28/2006

HOFFMAN WARNICK & D'ALESSANDRO, LLC
75 STATE STREET
14TH FLOOR
ALBANY, NY 12207

EXAMINER

CAMPBELL, JOSHUA D

ART UNIT PAPER NUMBER

2178

DATE MAILED: 11/28/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

NOV 28 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/709,433
Filing Date: November 13, 2000
Appellant(s): STEWART ET AL.

John W. LaBatt
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 5, 2006 appealing from the Office action mailed April 4, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,625,234	ADAMSKE et al.	9-2003
5,873,073	BRESNAN et al.	2-1999

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 12-27 and 29-33 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Adamske et al. (US Patent Number 6,625,234, filed on May 11, 1999).

Regarding independent claim 12 and dependent claims 32 and 33, Adamske discloses a method in which a user uses software on a client device to generate a print file based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). The user then uploads the print file to the server, this print file (PostScript) being capable of being directly printed by a printer (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). Adamske does not directly disclose in this embodiment that a preview is generated by the server and provided to the user based on the print file that was uploaded. However, Adamske discloses an alternate method in which the server generates a preview based on the print file and the configuration information and provides that preview to the user for display at the client device (column 5, line 64-column 7, line 15 of Adamske). It would have been obvious to one of ordinary skill in the art to combine the two methods of Adamske because it would have allowed the client system to do less work in the process.

Regarding dependent claim 13, Adamske discloses a method in which a print drive is installed on the client in order to generate the print file (column 5, line 64-column 7, line 15 of Adamske).

Regarding dependent claim 14, Adamske discloses a method in which a print driver is installed on the client and a print file is generated using the print driver, at which point the print file is uploaded to the server (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which the print driver necessary is automatically selected (column 5, line 64-column 7, line 15 of Adamske). Adamske does not disclose a method in which the print driver is listed. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have allowed to have listed the print driver of Adamske because it would have allowed the user to see the format type the print file would be in.

Regarding dependent claim 15, Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske).

Regarding dependent claim 16, Adamske discloses a method in which styles and printing options for the document are obtained and shown via the preview, which is then provided to the client (column 5, line 64-column 7, line 56 of Adamske).

Regarding independent claim 17, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). A plurality of copies is printed in accordance with a plurality of addresses that are obtained from the user (column 5, line 64-column 7, line 15 of Adamske). A coversheet and shipping label (memo) is customized for each address and recipient is printed; at point all parts are delivered to the delivery addresses (column 7, lines 16-56 of Adamske). Adamske does not disclose that the customized memo is obtained from the client. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have allowed the user to create the custom memo on the client rather than the server because it would have provided the user with the ability to create the memo with the same application as the document itself.

Regarding dependent claim 18, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske also discloses a method in which a

user interface is generated that may be web based (on the server) (column 5, line 64-column 7, line 15 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 5, line 64-column 7, line 56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). Adamske also discloses a method in which styles and printing options for the document are obtained and shown via the preview, which is then provided to the client (column 7, lines 16-56 of Adamske).

Regarding dependent claim 19, Adamske discloses a method in which a print drive is installed on the client in order to generate the print file (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a print driver is installed on the client and a print file is generated using the print driver, at which point the print file is uploaded to the server (column 5, line 64-column 7, line 15 of Adamske).

Adamske discloses a method in which the print driver necessary is automatically selected (column 5, line 64-column 7, line 15 of Adamske). Adamske does not disclose a method in which the print driver is listed. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have allowed to have listed the print driver of Adamske because it would have allowed the user to see the format type the print file would be in.

Regarding independent claim 20 and dependent claims 21-23, the claims incorporate substantially similar subject matter as claims 12-15. Thus, the claims are rejected along the same rationale as claims 12-15.

Regarding independent claim 24, Adamske discloses a method in which a user uses software on a client device to generate a print file based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). The user then uploads the print file to the server, this print file (PostScript) being capable of being directly printed by a printer (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). Adamske does not directly disclose in this embodiment that a preview is generated by the server and provided to the user based on the print file that was uploaded. However, Adamske discloses an alternate method in which the server generates a preview based on the print file and the configuration information and provides that preview to the user for display at the client device (column 5, line 64-column 7, line 15 of Adamske). It would have been obvious to one of ordinary skill in the art to combine the two methods of Adamske because it would have allowed the client system to do less work in the process.

Regarding dependent claim 25, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user

interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). A plurality of copies is printed in accordance with a plurality of addresses that are obtained from the user (column 5, line 64-column 7, line 15 of Adamske). A coversheet and shipping label (memo) is customized for each address and recipient is printed; at point all parts are delivered to the delivery addresses provided by the client (column 7, lines 16-56 of Adamske).

Regarding dependent claim 26, Adamske discloses a method in which payment information is obtained for the copy and the payment is processed using that information (column 6, line 58-column 7, line 15 of Adamske).

Regarding dependent claim 27, Adamske discloses a method in which the print driver generates the print file and an upload manager communicates the file to the server (column 5, line 64-column 7, line 15 of Adamske).

Regarding independent claim 29, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-

56 of Adamske). The interface is provided to the user via the Internet for display (column 2, lines 4-60 of Adamske). A plurality of copies is printed in accordance with a plurality of addresses that are obtained from the user (column 5, line 64-column 7, line 15 of Adamske). A coversheet and shipping label (memo) is customized for each address and recipient is printed; at point all parts are delivered to the delivery addresses (column 7, lines 16-56 of Adamske). Adamske does not disclose that the customized memo is obtained from the client. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have allowed the user to create the custom memo on the client rather than the server because it would have provided the user with the ability to create the memo with the same application as the document itself.

Regarding dependent claim 30, Adamske discloses a method in which the document may be generated on the client and obtained from the client (column 5, line 64-column 7, line 15 of Adamske).

Regarding independent claim 31, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display

(column 2, lines 4-60 of Adamske). A plurality of copies is printed in accordance with a plurality of addresses that are obtained from the user (column 5, line 64-column 7, line 15 of Adamske). A coversheet and shipping label (memo) is customized for each address and recipient is printed; at point all parts are delivered to the delivery addresses (column 7, lines 16-56 of Adamske). Adamske does not disclose that the customized memo is obtained from the client. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have allowed the user to create the custom memo on the client rather than the server because it would have provided the user with the ability to create the memo with the same application as the document itself.

Claim 28 remains rejected under 35 U.S.C. 103(a) as being unpatentable over Adamske et al. (US Patent Number 6,625,234, filed on May 11, 1999) in view of Bresnan et al. (US Patent Number 5,873,073, issued on February 16, 1999 - IDS).

Regarding dependent claim 28, Adamske discloses a method in which a user uses software on a client device to generate a print file and uploads it to a server or a print file may be generated on a server based on a document and a print driver (column 5, line 64-column 7, line 15 of Adamske). Adamske discloses a method in which a user interface is generated that may be web based (on the server) (column 2, lines 4-60 of Adamske). The interface provides a preview section and a printing options section that allows a user to provide configuration information (i.e. style options) (column 7, lines 16-56 of Adamske). The interface is provided to the user via the Internet for display

(column 2, lines 4-60 of Adamske). Adamske also discloses a method in which styles and printing options for the document are obtained and shown via the preview that may be navigated by selecting portions (navigation area), which is then provided to the client (column 5, line 64-column 7, line 56 of Adamske). Adamske does not disclose a method in which an estimate area displays an estimate of price based on the configuration and print file. However, Bresnan discloses a method in which an estimate is generated based on configuration and the file itself and presented to the user (column 14, lines 10-58 of Bresnan). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Adamske with the method of Bresnan because it would have allowed the user to see the price prior to the billing process.

(10) Response to Argument

Regarding appellant's arguments on pages 7-9, regarding claim 12 and whether or not Adamske teaches "generating a preview of a configured copy of the document on the server based on the print file and configuration information," the examiner maintains that the rejection is proper. Adamske teaches that a converted printable electronic document is used by the server to generate a preview of how that converted printable electronic document will look when it is printed in hard-copy form (column 6, lines 1-3 and 12-15 of Adamske). The claim itself requires that the preview be "based on the print file and configuration information," this wording of the claim is broad and there is no clear requirement that the preview directly correspond to the print file and the

configuration, rather only that it be “based on” the combination of the print file and the configuration, which is clearly shown by Adamske. In Adamske, the converted printable electronic document that is used to generate the preview is the finished converted version of the original document, which meets the user’s configuration for what the final hard-copy document will look like (column 5, lines 46-column 6, lines 23 of Adamske). Thus, the print preview that is generated is clearly based on the print file and the configuration information, actually in the case of Adamske it corresponds to what the final hard-copy will look like, thus it is a direct representation of the combination of the print file and the configuration options of the print job. Adamske even teaches that the user may change the configuration of the document after viewing the preview and then generate a new preview to see the effect of the changes (column 3, line 64-column 4, line 2 of Adamske), thus clearly showing that the preview is based on the document and the configuration information.

The appellant argues that the user may select certain print options after the print preview is verified, however this has no relevance to the claim as it is written. Once again, the claim merely states that the preview be merely “based on” configuration information. In response to appellant’s argument that the references fail to show certain features of appellant’s invention, it is noted that the features upon which appellant relies (i.e., selecting all print options before previewing) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Nowhere does the claim clarify that every piece of

configuration information must be obtained and that every piece of configuration information must be incorporated into the preview, rather the claim takes a broad approach to incorporating the configuration information into the preview, which in this case forces the examiner to maintain the broadest reasonable interpretation of the claimed invention based on the specification. This interpretation as explained above provides a basis for maintaining the rejection of the claims based on the Adamske reference.

Regarding appellant's arguments on pages 9-11, regarding claim 17 and whether or not Adamske teaches "obtaining memo information for a plurality of delivery addresses, wherein the memo information is customized for a recipient at each of the plurality of delivery addresses; printing a customized memo based on the customized memo information for each of the plurality of delivery addresses," the examiner maintains that the rejection is proper. While the examiner appreciates that the appellant took the time to provide definitions for the word "memo", the examiner feels it necessary to point out that for the purposes of examination of the application the interpretation of the claims will be based on the specification. Thus, if in the specification the appellant provided a definition or example of how a term in the claims is to be interpreted then for the purposes of examination the term will be interpreted based on that definition. In this case the appellant's specification clearly states, "The system also allows the user to create a customized memo, i.e., package cover slip," (page 25, lines 15-16 of Appellant's specification). Adamske teaches that a coversheet and shipping label are customized for each address and recipient is printed; at point all parts are delivered to

the delivery addresses (column 7, lines 39-43 of Adamske). The teachings of the coversheet and a shipping label of Adamske clearly correspond to the claimed "customized memo" as is clearly defined by the appellant's own specification, thus validating the rejection of claim 17.

Regarding appellant's arguments on pages 11-15, regarding claim 20 and whether or not Adamske teaches "system software that generates a print file on a client based on the document and communicates the print file to a server, where the print file can be directly printed by a printer," the examiner maintains that the rejection is proper. Adamske discloses that the client computer may include the print driver program which is operable to perform the conversion of the electronic document and provide the preview of the document prior to uploading the document to the web server (column 6, lines 34-42 of Adamske). Adamske clearly states that this print driver program includes all of the functionality of the print preview software program which is defined in server based embodiment of Adamske's disclosure (column 6, lines 3-12 of Adamske). Adamske actually gives a specific example of the program, GNU's Ghostscript software could be the print preview software program. Thus, Adamske discloses that the print driver program residing on the client will have all of the functionality of GNU's Ghostscript software. The examiner has attached the three page white paper published on October 15, 1998, of the Ghostscript software program from GNU as appendix A of this examiner's answer. This document effectively clarifies the functionality of the print preview software program due to the fact that Adamske stated that the Ghostscript software program could be the program used, and lays to rest the arguments the

appellant has set forth based on the incorrect interpretation of the Adamske reference's disclosure and invention.

The appellant argues that the file forwarded from the client to the server as disclosed in Adamske is not capable of being directly printed on a printer, however the examiner disagrees. First of it is important to note that the claim itself states "wherein the print file can be directly printed by a printer." This limitation simply states that a printer, not a specific printer, literally any printer can print the file created on the client. This limitation also does not require the file to be in a specific format, just any format that is capable of being printed by a printer. Adamske discloses that the print preview created by the print preview software would be JPEG images which by definition can be printed by a printer or in the case of Adamske simply viewed by the user (column 6, lines 3-12 of Adamske). In addition to this, the Ghostscript software program has three main functions, display a postscript file, display a postscript file to decide if you really need to print it, and print a postscript file to a non-postscript printer (Appendix A, page 1, lines 6-9). The last of the main uses is the most important for the purposes of this argument. Once again, the output of the program is a graphical output, in the case of Ghostscript it is usually a page bitmap (Appendix A, page 1, lines 3-5) which is by definition printable by a printer, actually a bitmap is printable by all printers, unlike Postscript which is the printable format used by the appellant's invention (page 17, lines 14-15 of Appellant's specification) which is only printable on Postscript enabled printers. The last goal of Ghostscript actually is to make the file printable on all printers, including non-Postscript enabled printers. Thus, the idea that file disclosed by Adamske, which is

generated by the Ghostscript program, can not “ be directly printed by a printer,” is incorrect, especially when the claim is interpreted in the broadest reasonable interpretation allowed by the specification. The idea that the appellant’s invention allows the file to be printed on any printer is neither enabled by the specification nor claimed, which once again forces the examiner to maintain the rejection of the claims based on the broadest reasonable interpretation, which as explained above is properly rejected based upon the teachings of Adamske.

Regarding appellant’s arguments on pages 15-17, regarding whether or not there is proper motivation to combine the two embodiments of Adamske, the examiner maintains that the rejection is proper based on the fact that there is indeed proper motivation. The appellant argues that the examiner primarily relies on the second embodiment of Adamske in which the client performs a majority of the processing (page 16, lines 10-12 of Appellant’s arguments). The appellant then argues that the examiner proposes to modify the second embodiment with the teachings of the first embodiment in which a majority of the processing occurs on the server, rather than the client (page 16, lines 12-15 of Appellant’s arguments). The examiner agrees with these statements, the second embodiment of Adamske is viewed as the primary reference, while the first embodiment is viewed as the secondary reference, the motivation to combine these references being that it would have allowed the client system to do less work in the process. The appellant states that the first embodiment (in this case the secondary reference) provides a solution in which the processing load at the client is reduced from that of the second embodiment (page 16, lines 17-19 of Appellant’s arguments). The

appellant argues that for this reason, the combination of the references is not proper, which is completely incorrect. The appellant is arguing that because the secondary reference addressed the motivation cited by the examiner it teaches away from the proposed combinations, which is incorrect.

In order to make a properly combine the teachings of the two embodiments of Adamske, it is necessary to show what the primary reference (second embodiment – column 6, lines 24-57 of Adamske) teaches, then show what it does not teach and show how the secondary reference (first embodiment – column 4, line 61-column 6, line 23 of Adamske) teaches the limitations. Then it is necessary to provide a motivation for using the teachings of the secondary reference (first embodiment) in combination with the first reference to cure the deficiencies of the first reference. Thus, the fact that the secondary reference (first embodiment) teaches that moving some of the functionality to the server from the client would in fact reduce the processing load at the client, which the appellant agrees is shown (page 16, lines 17-19 of Appellant's arguments), provides exactly the motivation necessary to make this combination because this combination would provide an improvement to the first reference (second embodiment) which is reducing processing load at the client. Thus, the rejection based on the combination of the two embodiments of Adamske is proper in both merit and motivation.

All of the other arguments on pages 17-20 are in reference to the topics above, thus the rationale above can be used to respond to the similar arguments.

Art Unit: 2178

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Joshua D. Campbell



Conferees:


Stephen Hong, Supervisory Patent Examiner for Group Art Unit 2178


Heather Herndon, Supervisory Patent Examiner for Group Art Unit 2176

Introduction to GNU Ghostscript

Ghostscript is an interpreter for the PostScript language. A PostScript interpreter usually takes as input a set of graphics commands. The output is usually a page bitmap which is then sent to an output device such as a printer or display. PostScript is embedded in many printers.

Ghostscript has several main uses:

1. Display a PostScript file (avoid killing trees).
2. Display a PostScript file to decide if you really need to print it (reduce the number of trees killed).
3. Print a PostScript file to a non-PostScript printer (kill more trees).

An example of a very simple PostScript file is:

```
%!  
/Helvetica findfont 72 scalefont setfont  
72 72 moveto  
(Hello, world!) show  
showpage
```

The first line is a PostScript comment. It is ignored by the PostScript interpreter, but enables other programs to recognise that the file contains PostScript commands. The second line says to find the Helvetica font, scale it to 72 points high (1 inch) and then set it as the current font. The third line moves the current point to 1 inch in from the left and bottom edges of the page. The fourth line draws the text into the page bitmap at the current point. The final line causes the page bitmap to be output to paper or the display.

The information below assumes the use of GNU Ghostscript 3.33. Other versions of Ghostscript are similar. The Ghostscript program is typically found in `/usr/local/bin/gs`

Ghostscript needs some initialization files in `/usr/local/lib/ghostscript/3.33`

and some fonts in `/usr/local/lib/ghostscript/fonts`.

The documentation files are in `/usr/local/lib/ghostscript/3.33/doc`

and some example PostScript files are in `/usr/local/lib/ghostscript/3.33/examples`.

Displaying a PostScript file

Let's assume that you wish to display the above PostScript example and it is in a file named `example.ps` in the current directory. If Ghostscript has been correctly installed, then you would start Ghostscript using:

```
gs
```

If Ghostscript has not been correctly installed, you will need to specify the full path, for example:

```
/usr/local/bin/gs -I/usr/local/lib/ghostscript/3.33:/usr/local/lib/ghostscript/fonts
```

The `-I` option tells Ghostscript to look for its configuration files and fonts in the two directories listed. After Ghostscript starts it will display

```
GNU Ghostscript 3.33 (4/10/1995)
```

```
Copyright (C) 1995 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
```

```
This software comes with NO WARRANTY: see the file COPYING for details.
```

```
GS>
```

<http://web.archive.org/web/19981206061415/http://www.gnu.org/software/ghostscript/intro.html>

An X11 image window will appear. Ghostscript is now waiting for you to tell it what to do. Typing `(example.ps) run` will produce

```
GS>(example.ps) run
Loading NimbusSanL-Regu font from /usr/local/lib/ghostscript/fonts\n0190031.pfb...
1669824 366951 1300076 11892 0 done.
>>showpage, press <return> to continue<<
```

The text "Hello, world!" will appear near the bottom of the X11 image window. When you have finished viewing the image window, switch to the text window and press the `<return>` key. You will be returned to the `GS>` prompt. Type `quit` to close Ghostscript.

```
GS>quit
```

If the file is not in the current directory, then you will need to give a full path. For example,
`(/user/john/example.ps) run.`

So now you know how to display a PostScript file. Some example PostScript files that come with Ghostscript are `colorcir.ps`, `chess.ps` and `tiger.ps`. These should be in the `/usr/local/lib/ghostscript/3.33/examples` directory.

Printing a PostScript file

In the above display example, we did not specify which output device Ghostscript should use. Ghostscript used the default device, which happened to be the display. To print a PostScript file, we need to tell Ghostscript the name of the printer device. For an HP DeskJet 500 or an HP DeskJet Portable printer, this is `djet500`. We would instead start Ghostscript with:

```
gs -sDEVICE=djet500
```

The output will be sent to a scratch file. Ghostscript will come up with the `GS>` prompt again, we give it a file name, and after processing each page it will pause. This is probably undesirable. A better way to start Ghostscript is:

```
gs -sDEVICE=djet500 -dNOPAUSE example.ps -c quit
```

This tells Ghostscript not to pause after each `showpage`, to interpret the file `example.ps` and afterwards to exit. This is much more useful.

To get a list of available printer devices, start Ghostscript for displaying, then type:

```
GS>devicenames ==
```

The destination for the printed output can be specified on the command line. To output to the file `out.prn`, add `-sOutputFile="out.prn"` somewhere before the name of the PostScript file. To pipe the output to the `lpr` program, use `-sOutputFile="\\lpr queueName"`

More details

If you want more details about the Ghostscript command line options, see the file **use.txt** or **use.doc** distributed with Ghostscript. This should be in the directory `/usr/local/lib/ghostscript/3.33/doc`.

A better way

If you are like me, you get sick of typing the long commands above. A program called Ghostview and several derivatives exist for X11 on Unix or VMS. This allows you to display page 20 without having to display pages 1, 2, 3, 4, ... 20. You can also print selected pages.

This page was written by Russell Lang. Return to GNU's home page.

Please send FSF & GNU inquiries & questions to gnu@gnu.org. There are also other ways to contact the FSF.

Please send comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.

Copyright (C) 1997, 1998 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

Updated: 15 Oct 1998 devnull
